California State University Northridge

College of Engineering and Computer Science
Mechanical Engineering Department

Mechanical Engineering 309
Numerical Analysis of Engineering Systems

Spring 2014 Number: 15237 Instructor: Larry Caretto

Course Outline

Catalog Description

Prerequisites: Math 150B; ME 209 or Comp 106/L or ECE 206/L or CE 280/L. Features engineering problems which require the use of algorithms and numerical analysis to obtain a solution. Modern tools such as spreadsheets with embedded high level languages are used for analysis and code development. Program documentation which requires extensive use of computer-based technical writing skills with graphical presentations. Cross section of problems are selected from various branches of engineering.

Instruction information	
Name	Larry Caretto
Email address	lcaretto@csun.edu
Office	Jacaranda 3314
Phone	Call department office in emergency
Fax	818.677.7062
Office hours	Monday-Wednesday, 6-6:45 pm and Friday 7:45 to 8:45 am
Course Information	
Course number	15237
Class hours	Monday-Wednesday, 7 to 9:45 pm
Class location	Jacaranda 1126

Expanded Description

This course covers various topics in numerical analysis and their implementation in the Excel spreadsheet and in MATLAB software. The Excel implementations cover both the use of spreadsheet formulas and the Visual Basic macro language. Numerical analysis will cover basic methods for various problems and the cautions that are associated with their use. This is a second course in programming and students are expected to have a background in Visual Basic for Applications (VBA) or an equivalent programming course in another language. This course will build on that background to increase students' programming proficiency in VBA and its use with Excel. The MATLAB material will discuss MATLAB basics, the use of MATLAB functions, and the basics of MATLAB programming.

Text and references

Text: Singiresu S. Rao, *Applied Numerical Methods for Engineering and Science*, Prentice-Hall (Pearson Education), 2002.

Reference: John Walkenbach, *Excel VBA Programming for Dummies*, Wiley, 2013.

Reference: Rudra Patrap, *Getting Started with MATLAB. A Quick Introduction for Scientists and Engineers*, Oxford University Press, 2009.

https://www.mathworks.com/academia/student_center/tutorials/mltutorial_launchpad.html?confirmation_page is a set of online MATLAB tutorials free for students and faculty

Course Conduct

Course Objectives – As a result of taking this course students should

- be able to apply various techniques in numerical analysis including roots of equations, simultaneous linear equations, numerical integration and differentiation, polynomial interpolation,

linear regression (statistical curve fitting) and the numerical solution of ordinary differential equations.

- understand the kinds of errors, such as truncation errors and roundoff errors, that can occur in finite-difference methods.
- understand and be able to use the basics of matrix analysis
- be able to write programs in Visual Basic that can be used as functions or macros for Excel spreadsheets.
- be able to use MATLAB functions for solving numerical analysis problems and obtaining symbolic mathematical solutions of algebraic equations, integrals and differential equations.
- be familiar with MATLAB code for m-files and be able to write simple code for use in MATLAB function calls.

Relation to program outcomes – As part of the accreditation process, the BS degree program in mechanical engineering has a set of outcomes that students should achieve by the time that they graduate. This course is designed to contribute to the following program outcomes for the BSME degree program: (a) the ability to apply knowledge of mathematics, science and engineering, (e) the ability to formulate and solve engineering problems, (f) an understanding of professional and ethical responsibility, (g) an ability to communicate effectively, (k) an ability to use techniques and skills of modern engineering tools, and (n) familiarity with statistics and linear algebra.

Class Participation – Learning computer programming is difficult for many people and it can only be mastered by practice. Your learning in this course will be a combination of textbook material, laboratory exercises, lecture material and in-class discussion. Your active participation in class exercises and discussion will help you learn the subject matter.

Class sessions – Students are expected to attend each class and spend the complete time in each laboratory session. In addition, students will spend a significant amount of time outside of the lecture and laboratory developing their algorithms into code and preparing their assignments for submission. If you do not do this you will not be able to complete your laboratory assignments in the time available. The laboratory sessions provide you with the main opportunity to get help from the instructor and from your classmates.

Grading – Your grade in this course will be based on eight in-class, 30-minute quizzes, seven programming assignments, a midterm, a programming exam and a final. These assignments will prepare you for questions on the midterm and final examinations. The assignments and exams will be weighted as follows in computing the final grade:

Quizzes	20%
Programming Assignments	20%
Midterm Examination	15%
Programming Examination	20%
Final	25%

The translation of a final numerical score into a letter grade rests solely on the judgment of the instructor. The following criteria will be used for letter grades:

A: Student knows almost all of the course material and is able to apply it to new algorithm developments and programming tasks.

B: Student satisfies one, but not both, of the conditions for an A grade.

C: Student knows fundamentals of the course and is able to apply this knowledge to routine algorithm development and programming.

D: Student has learned some course material but is not able to apply all the fundamental points of the course.

F: Student has failed to demonstrate knowledge of the course material beyond a minimal level.

Plus/minus grading will be used in this course. A plus grade indicates that the criterion for a given grade has been clearly met, but the student performance does not begin to approach the requirements for the

next highest grade. A minus grade is given when the student performance does not quite meet the requirements for the grade, but the criterion for the next lower grade has been substantially exceeded.

There will be no make-up exams or quizzes. Students who miss the midterm will have their grade for the missed midterm based on their relative performance on the final exam. Students who do not take the final will receive an F in the course. Students who are unable to take the final for valid reasons (sickness, death, *etc*.) **must request, in writing,** an incomplete grade in the course. (University regulations do not allow an instructor to assign an incomplete grade without a signed request by the student.)

Plagiarism vs*. Collaboration* – Students often work together on assignments. This collaboration is helpful and encouraged. By working together, each of you can improve your learning of the subject. It is important that you ask anyone who helps you, including the instructor, why they tell you to write a particular line or lines in your code. Unless you understand why you are writing some line of code, you are not learning.

In this course two students may collaborate on programming assignments and prepare only one submission for grading. Students who do this are expected to contribute equally to the final written submission and to work together on all parts of the assignment so that they understand all parts of the assignment.

There is a difference between working together to learn the material and copying someone else's work (either from other students or from some web site) and passing it off as your own. In this course you are expected to submit work that you have done yourself (or, for programming assignments with one other student.) Submitting another person's work as your own is a violation of academic standards and University regulations. It is unethical behavior for people working in engineering and science or studying to work in these fields. Each student must submit his or her own work to pass the course.

Students who copy programs from some other source and submit them, as their own work will receive a zero grade for the assignment. Where two or more students submit nearly identical programs, each student involved will receive a zero grade for the assignment. Any student who does this more than once will receive an F grade in the course.

Identical solutions on exams, indicating copying, will result in an F grade in the course for both students involved. Students who are found cheating by submission of identical assignments or any other observations will be referred to the Office of the Dean of Students for disciplinary action.

Late assignments – There will be two deadline dates for assignments. Assignments that are not submitted by the first date may be submitted by the second date with a penalty of 30% of the maximum grade. No assignments will be accepted after the second date.

Add-drop policy – Students are expected to be familiar with the University regulations for adding and dropping classes. Students who find that they do not have enough time to prepare for this class or whose performance on the initial quizzes is poor should drop the class within the appropriate deadline. (See http://www.csun.edu/anr/soc/adjsched.html for campus policies; for Spring 2014, the last day to drop a class without approval is Friday, February 7.) Students who do not complete the course work and do not withdraw from the class will receive a grade of WU, denoting an unsatisfactory withdrawal. Such grades count the same as an F grade in the computation of students' grade point averages.

Changes – Students are responsible for all changes to this outline announced in class.

Schedule of lecture topics

The reading assignments shown below are prefixed by the first initial of the author of the required text and references: **R** for Rao, **W** for Walkenbach, and **P** for Pratap.

Date	Lecture Topic	Tasks	Reading
January 22	Introduction to the course. Overview of Visual Basic used with Excel spreadsheets. Data types, operators, assignment statements. Initial exercise in use of spreadsheets and VBA.		**W** Ch 3 & 7

Date	Lecture Topic	Tasks	Reading
January 27	Introduction to visual basic continued: flow control, if statements, count-controlled and conditional looping, arrays, strings, creating functions.		**W** Ch 10 & 20
January 29	Conclusion of VBA review. Introduction to MATLAB.		**P** pp 1-10
February 3	MATLAB scripts and functions. Use of symbolic mathematics in MATLAB. Plotting in MATLAB. Use of diary and publish commands	**Program 1 Due**	**P** pp 11-24
February 5	Introduction to numerical analysis. Representation of numbers, real and integer data type representations, definitions of error, and introduction to finding roots of equations.	**Quiz 1**	**R** ch 1
February 10	Roots of equations. Use of the Goal Seek and Solver functions of Excel. Introduction to methods for finding roots. Searching to bracket a root. Using the MATLAB function, fzero.		**R** pp 58-64 **P** pp 106, 168-170
February 12	Discussion of various methods for roots of equations: bisection, regula-falsi, secant method, Newton's method. Library routines for finding roots of nonlinear equation systems. Roots of polynomial equations. MATLAB's roots function for roots of polynomials.	**Quiz 2**	**R** pp 88-100 **P** pp 170-171
February 17	Basics of matrix analysis. Use of matrices to represent systems of simultaneous linear equations and their solutions.	**Program 2 Due**	Notes
February 19	Introduction to simultaneous linear equations. General notation for equations. Matrix notation. Existence and uniqueness of solutions. Solutions by Excel and MATLAB functions.	**Quiz 3**	**R** pp 141-159
February 24	Gaussian elimination. Writing general software for solution of simultaneous linear equations		**R** pp159-172
February 26	Other approaches related to Gaussian elimination. LU decomposition, Gauss-Jordan. Finding the inverse of a matrix.	**Quiz 4**	**R** pp 172-181
March 3	Iterative approaches to simultaneous linear equations; Jacobi, Gauss-Seidel and successive overrelaxation	**Program 3 Due**	**R** pp 172-181
March 5	Use of Excel solver for optimization problems and for finding roots of simultaneous non-linear equations.	**Quiz 5**	
March 10	Use of Taylor series to derive finite-difference approximations, truncation error, roundoff error, interaction between the two types of error. Order of the error.		Notes
March 12	Polynomial interpolation. General interpolation polynomial.	**Quiz 6**	**R** pp 364-370
March 17	Interpolation with fixed spacing. Inverse interpolation.	**Program 4 Due**	**R** pp 370-383
March 19	Spline fits. Practical polynomial interpolation: piecewise polynomials *versus* higher-order polynomials.		**R** pp 387-399
March 24	Review for midterm.		
March 26	**Midterm Exam**		
March 31	**Cesar Chavez Holiday – No Class**		
April 2	Contrasting polynomial interpolation with statistical data fitting (regression analysis). Functions for data fitting in Excel (LINEST) and MATLAB.		**R** pp 399-422 **P** §§ 5.2&5.3
April 7 and 9	**Spring Break – No Class**		

Date	Lecture Topic	Tasks	Reading
April 14	Introduction to numerical differentiation and integration. Formulas for calculating derivatives by numerical analysis. Effect of order.	**Program 5 Due**	**R** pp 500-529
April 16	Numerical integration by trapezoidal rule and Simpson's rule. Error in the numerical integration. Programming techniques for Simpson's rule.	**Quiz 7**	**R** pp 560-587
April 21	Miscellaneous issues in numerical integration: uneven spacing, multiple integrals, indefinite integrals, Gaussian integration. MATLAB functions for numerical and symbolic integration.		**R** pp 605-607 **P** §§ 5.4
April 23	Introduction to numerical solution of ordinary differential equations. The initial value problem.	**Quiz 8**	**R** pp 631-642
April 28	Euler, modified Euler and Runge-Kutta methods – algorithms and error analysis of the algorithms. Error propagation.	**Program 6 Due**	**R** pp 647-665
April 30	Integrating systems of ordinary differential equations. The need for error control and adaptive step sizes.		**R** pp 683-691
May 5	Multistep methods, predictor corrector, error analysis for adjusting step size. Boundary-value problems. Solution by shoot-and-try methods. Finite-difference methods. Review for final.	**Program 7 Due**	**R** pp 665-683, 723-752
May 7	**Programming exam**		
May 12	**Final Exam, Monday May 12, 8 to 10 pm**		

References for VBA and numerical analysis

See the course web site http://www.csun.edu/~lcaretto/me309/sites.htm for references including a printable list of references.